Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2023"

June 14–16, 2023

Incremental Topic Modeling for Scientific Trend Topics Extraction

Nikolai Gerasimenko
Sberbank, MSU Institute for Artificial Intelligence
nikgerasimenko@gmail.com

Alexander Chernyavskiy
National Research University Higher School of Economics
alschernyavskiy@gmail.com

Maria Nikiforova
Sberbank
labenzom@gmail.com

Anastasia Ianina
Moscow Institute of Physics and Techonology (MIPT)
yanina@phystech.edu

Konstantin Vorontsov
MSU Institute for Artificial Intelligence, MIPT
vokov@forecsys.ru

Abstract

Rapid growth of scientific publications and intensive emergence of new directions and approaches poses a challenge to the scientific community to identify trends in a timely and automatic manner. We denote trend as a semantically homogeneous theme that is characterized by a lexical kernel steadily evolving in time and a sharp, often exponential, increase in the number of publications. In this paper, we investigate recent topic modeling approaches to accurately extract trending topics at an early stage. In particular, we customize the standard ARTM-based approach and propose a novel incremental training technique which helps the model to operate on data in real-time. We further create the Artificial Intelligence Trends Dataset (AITD) that contains a collection of early-stage articles and a set of key collocations for each trend. The conducted experiments demonstrate that the suggested ARTM-based approach outperforms the classic PLSA, LDA models and a neural approach based on BERT representations. Our models and dataset are open for research purposes.

Keywords: Topic modeling, Trend Extraction, Additive Regularization of Topic Models, Incremental Topic Modeling.
DOI: 10.28995/2075-7182-2023-22-88-103

Инкрементальная тематическая модель для выделения научных тематических трендов

Аннотация

Быстрый рост количества научных публикаций и интенсивное внедрение новых направлений и подходов исследований значительно усложняет проблему автоматического выделения научных трендов. Мы определяем тренд как семантически однородную тему, которая характеризуется постепенно эволюционирующим лексическим ядром, а также резким, часто экспоненциальным, скачком количества публикаций в начале развития тренда. В этой статье мы применяем тематическое моделирование для выделения трендовых тем на раннем этапе их развития. Визоизменив стандартный подход АРТМ, мы создали новую технику инкрементального обучения тематических моделей, которая может дообучаться с использованием актуальных статей в режиме реального времени. Также мы представляем датасет трендов по искусственному интеллекту (Artificial Intelligence Trends Dataset, AITD), который содержит коллекцию статей и набор ключевых слов для каждого тренда. Проведенные эксперименты показывают, что предложенный подход на основе АРТМ превосходит классические алгоритмы (PLSA, LDA) и нейронные подходы на основе BERT. Наши модели и датасет доступны для исследовательских целей.

Ключевые слова: тематическое моделирование, выделение трендов, аддитивная регуляризация тематических моделей, инкрементальное тематическое моделирование.

1 Introduction

The rapid growth of scientific publications, journals, and conferences makes it effortful to reconstruct a complete purview of specific subject areas. Nowadays, people have to keep track of numerous emerging areas and domains, for which the global scientific importance is not always explicit at the first sight. In this regard, more attention is paid to methods that solve the research trend identification task (Ho et al., 2014; Rotolo et al., 2015; Prabhakaran et al., 2016; Färber and Jatowt, 2019; Uban et al., 2021).

In this study, we consider the task of trend-like topic detection in real-time. The resulting topics should comply with the following conditions:

1. They should contain as many trending topics as possible. Here, we apply the definition of a trend proposed by (Kontostathis et al., 2004), where the emerging trend is defined as a topic, interest to which was strongly increasing in a particular time interval.
2. Trend-like topics should be identified as early as possible by the time they appear.
3. Each topic should be semantically homogeneous and impartible. This formulation imposes specific restrictions.

In our experiments, we extract trending publications in the field of Artificial Intelligence (AI), but the proposed approach can be applied to other scientific fields a well.

Let us consider an example with the ELECTRA model (Clark et al., 2020). It immediately aroused great interest among the scientific community and began to be actively used in various applications. Hence, more than 200 articles referring to it had already been published in 2020 alone. This certainly conforms to our definition of a trend, and our goal is to build a system that will also be able to highlight such trends as early as possible.

It should be emphasized that the trend can be not only a *model* but also a *task* (like the fact-checking task) or a *method* (like Dropout or AdamW). Moreover, we do not aim to utilize models only for retrospective analysis and highlight the main trends in the past. Thus, we set the problem statement in such a way that the system is allowed to make predictions into the future, that is, to distinguish research areas that are currently developing most actively.

In order for the final model to operate in real-time, we suggest incremental training. At each timestamp, we aim to generate new topics as distant as possible from existing ones, which is not implied a priori in some topic models. Further, many current topic modeling approaches have issues associated with the dilution of topics and terms, and the decorrelation of terms. To overcome these and other similar problems, we apply a topic model with additive regularization, namely ARTM (Vorontsov and Potapenko, 2015). Moreover, we offer several ways to customize it that contributes to achieve the best quality.

Despite active research in the field, there is no single quality metric for comparing trend detection models. Thus, we propose our intuitive metric in accordance with the assigned task.

Apart from that, we create a special expertly assembled dataset for comparison, which we issue in the public domain. We called it Artificial Intelligence Trends Dataset (AITD).

Our contributions can be summarized as follows:

- We propose the incremental mechanism of ARTM training to detect trend topics in real-time.
- We propose the novel ARTM-based approach that outperforms popular neural network and topic modeling approaches in the task of early trend detection.
- We create a specialized dataset to validate trend topic detection approaches, which we release for the research community.
- We make our approach and the created dataset open releasing code and the data there: `https://drive.google.com/file/d/1ueb9OgTdeITkOCly7doKO4KwL7G_YjT_/view?usp=share_link`.

2 Related Work

Trend detection systems generally can be divided into two groups: semi-auto and auto approaches. We investigate only approaches that do not require human interaction.

Generally, automatic detection of trends involves two stages: topic detection (or identification) and topic evolution (with emerging trend classification). The first stage is needed to construct the set of topics from which the trends will be selected. The following types of approaches can be distinguished: statistical, knowledge-based, and hybrid. Statistical approaches use only the given textual context without any additional meta-information. Various models have been already investigated in this direction: topic modeling (Prabhakaran et al., 2016; Uban et al., 2021; Krivenko and Vasilyev, 2009), clustering approaches

(Mei and Zhai, 2005; Behpour et al., 2021), and so forth. Among the aforementioned models a sequential variant of LSI (Krivenko and Vasilyev, 2009) is the approach most similar to ours in terms of problem formulation. Apart from that, other models utilize information from knowledge bases like the web (Roy et al., 2002) or citation graphs (Erten et al., 2004; Chang and Blei, 2010). Hybrid approaches (Jo et al., 2007; He et al., 2009; Ma et al., 2010) combine term-based topic detection and co-citation/co-authorship graph analysis.

There also has been some research on neural approaches for topic modeling (e.g. Transformer-based) (Grootendorst, 2020; Angelov, 2020). However, these approaches are not directly applicable due to the specifics of our collection, namely the length of the full texts of articles. For instance, the BERT model (Devlin et al., 2018a) has a limit on the length of input sequences of 512 tokens. Thus, it is needed to use either aggregation or additional models for text summarization to construct embeddings of entire texts. Nonetheless, we use the BERTopic model (Grootendorst, 2020) for comparison and show its inefficiency compared to our approach.

Topic evolution is utilized to consider topic emergence in time. Here, some approaches use custom metrics based on the topic characteristics (Ho et al., 2014; Prabhakaran et al., 2016; Grosso et al., 2017; Färber and Jatowt, 2019; Behpour et al., 2021). Another category of approaches considers citations-based metrics. In this way, (Le et al., 2006) proposed to use various temporal citation-based features to evaluate the growth in interest and utility of topics over time. In this work, we do not investigate classification of topics into trends and non-trends and mainly focus on the first part of the trend extraction pipeline. However, experiments with the trend evolution analysis are a subject for the further research.

To track topic emergence in real time, we investigate incremental topic models. Some researchers suggested online techniques for LDA (Canini et al., 2009; Hoffman et al., 2010). Nevertheless, due to the qualitative limitations of LDA-based approaches which are confirmed by our experiments, we use the ARTM model (Vorontsov and Potapenko, 2015) and propose a method of its incremental training. Our incremental mechanism is based on trend keywords detection. Similar to our approach, (Färber and Jatowt, 2019) proposed a method to estimate the impact index of keywords but did not integrate it into the trend detection pipeline.

Later (Sivanandham et al., 2021) proposed a model for analyzing research trends using topic modeling (LDA) and vector auto regression. On the contrary, (Lee et al., 2021) applied language modeling (BERT) and t-SNE algorithm for future prediction of growth potential of technologies.

The most recent studies mostly focus on neural topic models bridging the gap between probabilistic dynamic topic models based on matrix factorization techniques and the power of large language models. For example, Aligned Neural Topic Model (ANTM) (Rahimi et al., 2023) uses document embeddings to compute clusters of semantically similar documents at different periods of time and then aligns document clusters to represent their evolution. ANTM outperforms models Dynamic Embedded Topic Models (Dieng et al., 2019; Dieng, 2020) and significantly improves topic coherence and diversity over other existing dynamic neural topic models (e.g. BERTopic (Grootendorst, 2020)).

Another interesting research direction that can be easily applied for trend extraction is Graph Neural Networks. Such approaches preserve document dynamics and network adjacency by saving document relatedness via graph edges. For example, (Liang et al., 2023) fuses the graph topology structure and the document embeddings, while (Zhang and Lauw, 2022) proposes two neural topic models aimed at learning unified topic distributions that incorporate both document dynamics and network structure.

3 Trend Topic Detection

3.1 Task Definition

We consider the task of trending topic detection in real-time. In order to experiment not only with models based on matrix factorization but also with other popular approaches (e.g. clustering-based), we suggest to reduce the topic detection task to a search problem. Broadly speaking, we have a query for each topic (a topic name), and the goal is to get relevant lists of terms and documents associated with it. In our case, the queries are hidden, but we can still solve the recommendation task for them. Thus, the system should return ranked lists of per-topic documents and words for each predefined timestamp.

3.2 Approach

To obtain real-time predictions and reduce training time, we suggest incremental training of the topic model. The model leverages an incremental approach to create new topics based on words and collocations appearing in the last time interval, which contributes to more accurate trend extraction. The incremental model solves two subtasks: choosing the number of new topics, initializing new topics and adjusting them later.

We chose ARTM (Vorontsov and Potapenko, 2015) as the base model since it allows to build multi-objective models adding multiple criteria in a form of regularizers.

Base Model The ARTM model, in contrast to the LDA model that considers only a Dirichlet regularizer, allows to regard nonstandard important regularizers: smoothing and thinning of distributions of terms and topics, decorrelating distributions of terms in topics. Thus, we chose it as the base model for our topic modeling approach.

Initialization Let D be a collection of documents and W be a dictionary of words. After a new collection of documents D' appears, the model considers a set of emerging words W' and updates current topics T by adding new topics T' to it.

Generally, topic modeling approaches operate with matrices Φ and Θ representing word-topic and topic-document distributions respectively.

We suggest an incremental update to each of them. So, we initialize the matrices Φ_{n+1} and Θ_{n+1} in the current step using the matrices Φ_n and Θ_n from the previous step. More specifically, we copy Φ_n to the $\{W \leftrightarrow T\}$ submatrix of the matrix Φ_{n+1}, and Θ_n — to the $\{T \leftrightarrow D\}$ submatrix of the matrix Θ_{n+1}. All other values are filled according to the uniform distribution.

Number of New Topics The number of new topics for updating can be chosen in various ways (based on new documents collection, new terms or some combination of them). Here, we consider two of them: (*i*) a base straightforward approach that adds a fixed number of topics, (*ii*) a customized approach based on the emerging trend vocabulary V that is constructed based on impact scores similar to scores from (Färber and Jatowt, 2019).

In the base approach, we firstly count the mean value of terms related to each topic. This can be done by training one topic model for the first timestamp. Next, a new topic is created when the corresponding number of new terms appears in the vocabulary of key terms. This is because we are changing the current vocabulary to maintain a fixed size. Thus, some of collocations removed or added over time.

In the custom approach, the emerging trend vocabulary V consists of terms that have become much more commonly used compared to the moment of the last update of the topic model.

Let $w \in W \cup W'$ be a word from the current corpus. At the current timestamp, this word is added to V if it appears in at least $mindf$ documents and it satisfies the trend condition:

$$\frac{\text{tf}_{\text{new}} - \text{tf}_{\text{old}}}{\text{tf}_{\text{old}}} > \alpha \tag{1}$$

Here, tf_{old} is the count of the occurrence of w in documents D, and tf_{new} is the count of the occurrence of w in $D \cup D'$. $\alpha \in (0, 1)$ is a regulation hyper-parameter that sets the degree of increase in the occurrence of words to classify them as trending.

$$|T'| = |T_{\text{start}}| + \left\lfloor \frac{|V|}{\beta} \right\rfloor \tag{2}$$

In (2), T_{start} determines the number of topics at the initial timestamp, $\beta \in \mathbb{N}$ limits the number of added topics, and $\lfloor \cdot \rfloor$ denotes an integer part.

In other works the strategy of choosing number of topics in topic models include approaches based on simple heuristics and grid search (Ianina and Vorontsov, 2019; Ianina and Vorontsov, 2020), minimax optimal guarantees (Bing et al., 2020) or Bayesian approach and GNNs (Loureiro et al., 2023). Detailed comparison between methods of choosing the right number of topics is beyond the scope of this work.

Training Document Collections The result is also affected by the set of documents used for the model retraining at each timestamp for update: there are options to take either all documents in the history, or only new ones, or some intermediate option (with overlapping).

Formally, we have several options for the training document collection $\hat{D}$ at each step t:

$$\hat{D} = \begin{cases} D_t \\ D_t \setminus D_{t-1} \\ D_t \setminus D_{t-k} & \text{for } 1 < k < t \end{cases} \tag{3}$$

All these options affect the training time, and the second allows us to learn in real-time. In our experiments, we analyze these options in terms of quality and efficiency for our task.

Topic Models as Recommendation Systems To solve the recommendation task, we leverage probability scores from Φ and Θ to rank documents and words in the most appropriate way for each topic. A higher probability indicates that the model considers the document or word to be more important.

3.3 Evaluation

We propose a matching stage to map the labeled trends to the detected topics. At each iteration of the additional training of the incremental model, the search for the best topic for each trend is performed as follows.

Let D_{trend} and W_{trend} be the labeled sets of documents and words associated with the given trend respectively. Apart from that, we consider "golden" set of topic names S_{trend} . Here, S_{trend} contains from one to three synonymous collocations, each of which can be used as the trend name. At the output stage of the model each topic is represented by two ranked lists denoted as D_{topic} and W_{topic}. Also, we define $S_{\text{topic}} := W_{\text{topic}}$.

To perform matching, we firstly calculate three Recall@k based metrics:

$$\text{XRecall@k} = \frac{|X_{\text{topic}}[\text{:k}] \cap X_{\text{trend}}|}{k} \tag{4}$$

Here, $X[\text{:m}]$ denotes first m elements of the list X, where X is W, D or S respectively. We use three different values of the parameter k for documents, words and topic names, which are denoted as k_D, k_W and $k_S \leq k_W$ respectively.

We combine DRecall@k, WRecall@k and SRecall@k scores to estimate the relevance of the selected topic to the selected trend. We consider the trend to be detected once it has been matched with one of the extracted topics.

Since our goal is to minimize time delay for the trend detection, the final quality metric is the average number of days (or timestamps) that elapsed from the inception of a trend to its detection by the model. In our case, the inception date is the date of the earliest publication from the dataset.

4 Dataset

4.1 Background

To validate topic models, we collected a dataset of scientific trends. The closest work to us is the TRENDNERT benchmark proposed by (Moiseeva and Schütze, 2020), where the first public baseline for detecting (down)trends was presented. The dataset was constructed from a subset of papers published from 2000 to 2015.

Despite the large volume, the TRENDNERT benchmark has several drawbacks. Firstly, due to the fact that stratification was used for documents selection, some key papers that had a high impact on the trends at the beginning of their evolution could be lost. Secondly, the trends presented in this benchmark can be obtained by mapping internal identifiers proposed by the authors of the paper to an identifier from the Semantic Scholar database. However, we found this mapping outdated and results cannot be 100% replicated.

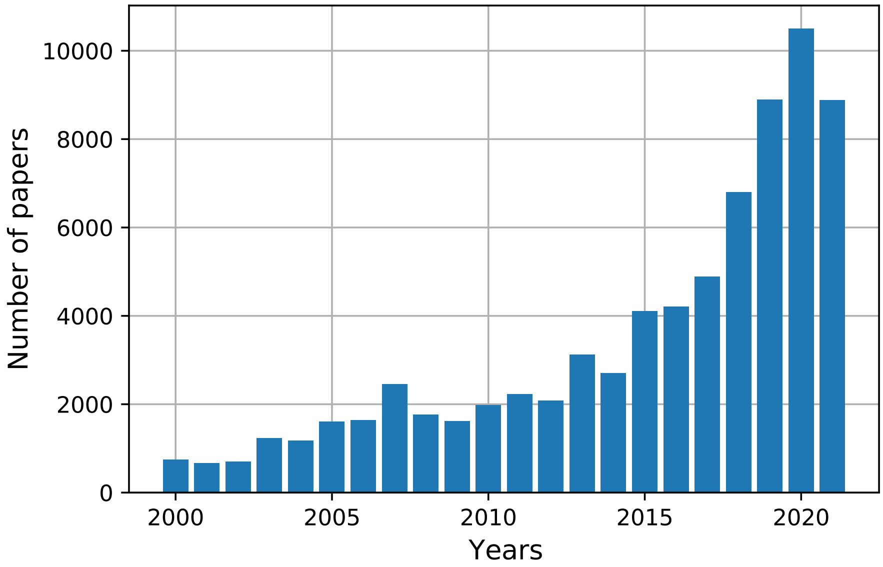

Figure 1: Papers distribution from selected conferences.

To overcome the drawbacks listed above, we present a new dataset, namely `AITD`. It focuses on trends in the Artificial Intelligence field across 2009-2021 years.

4.2 Data Sources

We used the part of Semantic Scholar Open Research Corpus as the main source of scientific publications, namely the Computer Science section (∼18M articles) with publications from 2000 to 2021. To filter the dataset, we considered only publications from 11 conferences that were selected based on data of top venues of Google Scholar[1] (Artificial Intelligence, Computational Linguistic, Computer Vision & Pattern Recognition sections were chosen) and h-index exceeding 100. Further, we filtered publications that did not contain any information about the corresponding conference name or the year of publication.

Figure 1 demonstrates the number of papers published by years from 2000 to 2021. It can be seen that almost every year the number of papers increases, and most of them were published relatively recently.

Final list of conferences and number of papers from each of it presented in Table 1. It demonstrates that the most of papers were presented on the computer vision CVPR conference. Apart from that, most of the dataset (more than 40%) is made up of articles from general conferences: NeurlIPS, AAAI, IJCAI and ICML.

We enriched our dataset by adding information from the arXiv dataset[2], and updated years for some publications. Thus, we solve the problem of data leakage for trend detection. That is, we exclude the situation when the article was first published on the arXiv site and became available to the scientific community and only after some time appeared in the proceedings of some conference.

Eventually, our dataset contains the following attributes: the paper id on Semantic Scholar, the title, authors' ids, venue, ids of publications it refers to, ids of papers that refer to it, the date of publication on arXiv, and the date of the conference. For the dataset construction, we utilized SciPDF Parser4 and PyMuPDF5 to extract text layer from the downloaded PDF files. We extracted collocations using the TopMine (El-Kishky et al., 2014) algorithm. To avoid "looking into the future" data leaks, the dataset was divided into subsets by two-week time intervals from March 2000 to December 2021.

[1] `http://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng`

[2] `https://www.kaggle.com/Cornell-University/arxiv`

Conference	Number of papers	% in the dataset
CVPR	11547	15.61
NeurIPS	11423	15.45
AAAI	9408	12.72
IJCAI	7523	10.17
ICML	7143	9.66
ACL	6752	9.13
ICCV	4976	6.73
EMNLP	4855	6.56
ECCV	4030	5.45
NAACL	3192	4.32
ICLR	3110	4.21

Table 1: Distribution of papers in dataset per conference.

4.3 Labeling

Trends Generation To prepare the validation dataset, we used the reference graph from the Semantic Scholar dataset. Initially, we generated manually 91 trends (for “model”, “method”, and “task” types) in the field of Machine Learning and Artificial Intelligence (e.g. CNN, RNN, BERT). For each of them, we found a paper with which this trend began or revived, as we call it “first story”. This concept can be illustrated with Fig.**??**: the evolution of each trend is described with a number of relevant papers being published at the selected period of time. There may be clear upword trends (e.g. "CNN" in Fig. 2) or trends with more complicated evolutional paths (e.g. "PCA" in Fig. 3)

Further, for each trend, we expertly selected at least 10 relevant publications based on the citation graph and used collocations. For the chosen papers, we analyzed the most frequent collocations and selected only those that are directly related to the topic of the trend (more than 5 keywords per trend).

To this end, we firstly collected a list of machine learning concepts frequently mentioned in scientific publications. We considered the exponential growth of mentions from some point in time as the condition for a topic to be a trend. The year starting from which the mentions rapidly grew was considered as trend start date. The dataset is not balanced and the maximum number of trends (more then 17) appeared in 2015. The distribution has light tails with relatively few trends (less than 5).

Papers and Keywords Selection After the first paper of the trend is found (the paper that created or re-invented the trend topic), we select related articles for each trend. The following conditions were used: (1) the selected articles should be directly related to the trend; (2) the articles should be published no later than two years after the first paper of the trend. For each trend, at least 10 articles that satisfy the conditions were selected. Further, collocations were selected from those papers to create keyword lists (at least 10 keywords for each trend).

Trend Names Labeling The last step was to choose alternative names for the trends based on the general knowledge or keywords. All the names were selected from the fixed collocations vocabulary.

Final Dataset Thus, we collected the dataset with the following structure: trend name, a subset of papers related to the trend, trend keywords, possible trend names. During the dataset construction, we utilized SciPDF Parser[3] and PyMuPDF[4] to extract text layer from downloaded PDF files. Unparsed articles are not further considered.

[3] https://github.com/titipata/scipdf_parser

[4] https://github.com/pymupdf/PyMuPDF

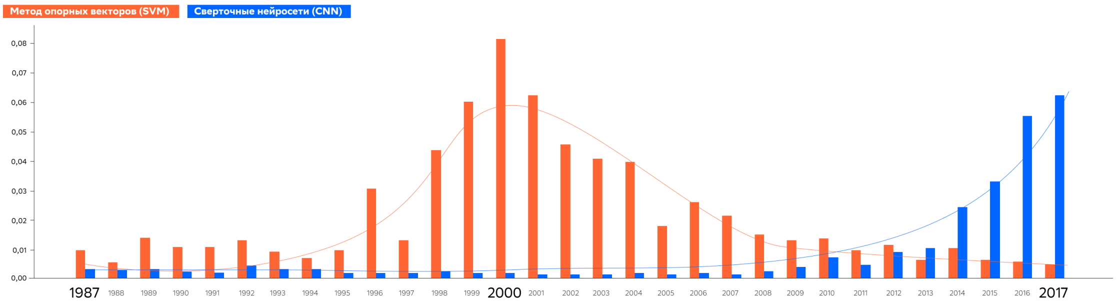

Figure 2: The example of two trends (CNN and SVM) evolving in time.

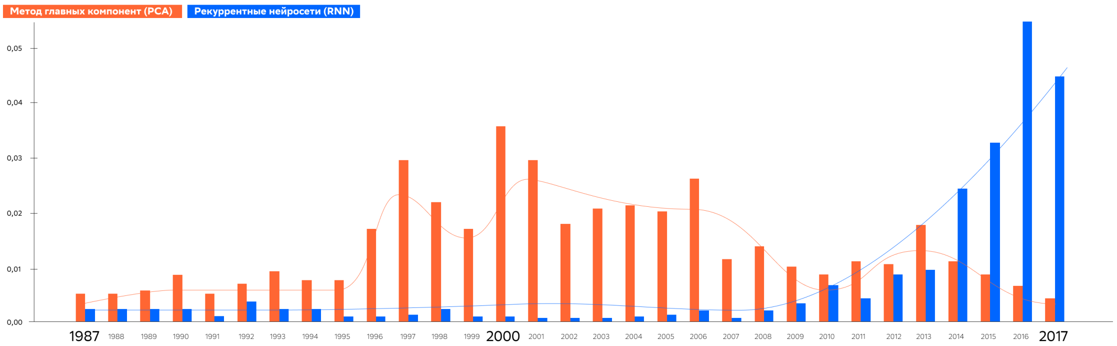

Figure 3: The example of two trends (PCA and RNN) evolving in time.

We extracted collocations using the TopMine (El-Kishky et al., 2014) algorithm. To avoid looking into the future, the dataset was divided into subsets by two-week time intervals from March 2000 to December 2021. We passed these batches to TopMine to extract collocations with maximal length of 5 words.

The dataset is publicly available here: `https://drive.google.com/file/d/1ueb9OgTdeITkOCly7doKO4KwL7G_YjT_/view`.

5 Experiments

5.1 Implementation Details

We trained our model on a sequence of time periods. We chose these periods in such a way that each period was at least two weeks long and contained at least 1000 published documents. As a result, we obtained 82 periods for training.

The open-source BigARTM library (Vorontsov et al., 2015) was used to train PLSA (Hoffman, 1999), LDA (Blei et al., 2003) and ARTM (Vorontsov and Potapenko, 2015) models. For ARTM, we used the regularizer named Decorrelator Φ that contributes to the decorrelation of columns in the Φ matrix. The regularization coefficient was set to 0.2. We also used the SmoothSparse Θ regularizer for which regularization coefficient was set to -1.

5.2 Models

Code for experiments was written on Python 3.

We conducted our experiments for sequence of timestamps, updating every 2 weeks, if at least 1000 new documents had been published in this period. For our dataset we got 82 timestamps and for each of them the batch of documents was created.

The open-source BigARTM library (Vorontsov et al., 2015) was used to train PLSA, LDA and ARTM models. In the case of the ARTM model, we used the regularizer named Decorrelator Φ that contributes to the decorrelation of columns in the Φ matrix. The regularization coefficient was set to 0.2. We also used the SmoothSparse Θ regularizer and regularization coefficient was set to -1.

In the process of the incremental learning when the sparsity of matrix Φ achieved 0.9 the Decorrelator Φ turned off. Similarly, when the sparsity of matrix Θ achieved 0.9 SmoothSparse Θ turned off. If sparsity drops below 0.9, then regularizers turn back on. We also used the same procedure with LDA model, because Dirichlet Regularizers had poor sparse effect on Φ and Θ matrices.

In the incremental learning process we used early stopping criteria. If within three passes the topic models perplection changes by less than 5% over subcollection of current incremental steps, then the learning process ends and the model proceeds to a new incremental step. Also model goes to the next incremental step upon reaching 24 collection passes on a incremental steps subset.

5.3 Baselines

We consider several baselines to compare our solution with.

Probabilistic Latent Semantic Allocation (PLSA) PLSA (Hoffman, 1999) is historically the first probabilistic topic model. Within PLSA one finds an approximate representation of counter matrix $F = (\frac{n_{dw}}{n_d})_{W \times D}$ (n_{dw} and n_d are counters of occurrences of term w in document d and overall number of terms in document d respectively) into a product of two unknown matrices — matrix Φ of term probabilities for the topics and matrix Θ of topic probabilities for the documents. In ARTM formulation PLSA corresponds to the model with no regularizers.

For consistency, in our experiments we used implementation of PLSA from BigARTM library (Vorontsov et al., 2015). The number of topics was chosen to be 200.

Latent Dirichlet Allocation (LDA) LDA (Blei et al., 2003) is a three-level hierarchical Bayesian model, in which documents are represented as random mixtures over latent topics, where each topic is characterized by distribution over words. Following the formulation of the problem from PLSA, in LDA parameters Φ and Θ are constrained by an assumption that vectors ϕ_t and θ_d are drawn from Dirichlet distributions with hyperparameters $\beta = (\beta_w)_{w \in W}$ and $\alpha = (\alpha_t)_{t \in T}$ respectively. In ARTM formulation LDA corresponds to the model with two regularizers that force an assumption that Φ and Θ columns are generated from Dirichlet distribution with hyperparameter β and α respectively. Following formulas represent corresponding regularizers within LDA model:

$$R(\Phi) = \sum_{t \in T} \sum_{w \in W} (\beta - 1) \ln \phi_{wt} \to \max$$

$$R(\Theta) = \sum_{d \in D} \sum_{t \in T} (\alpha - 1) \ln \theta_{td} \to \max$$

We used LDA implementation from BigARTM library (Vorontsov et al., 2015). Hyperparameters for LDA model were set to default values for symmetric Dirichlet distribution: $\alpha = \frac{1}{|T|}$, $\beta = \frac{1}{|T|}$, whete $|T|$ is the number of topics.

ARTM with decorrelation regularizer Another baseline is ARTM model with just one regulizer: decorrelation of matrix Φ. It is used to determine the lexical kernel of each topic which distinguishes it from the other topics. It minimizes covariations between columns of the Φ matrix:

$$R(\Phi) = \tau \sum_{t \in T} \sum_{s \in T \backslash t} \sum_{w \in W} \phi_{wt} \phi_{ws} \to \max$$

In our experiments coefficient of regularization τ is equal to 0.2.

Statistic	***Config1***				***Config2***				***Config3***			
	PLSA	**LDA**	**BERTopic**	**ARTM**	**PLSA**	**LDA**	**BERTopic**	**ARTM**	**PLSA**	**LDA**	**BERTopic**	**ARTM**
mean	295	268	**76**	181	526	519	685	586	731	761	608	**538**
min	1	1	**0**	**0**	**4**	**4**	10	**4**	**4**	11	10	11
25%	45	45	**14**	22	38	**23**	176	52	190	152	176	**110**
50%	126	114	**45**	56	443	**361**	484	476	556	504	**420**	479
75%	282	249	**95**	120	847	**827**	966	867	1074	1156	989	**761**
max	2907	2659	**871**	3433	**1921**	2273	2319	2711	2907	2659	2131	**1949**
# extracted	70	76	**90**	74	51	**53**	36	**53**	34	**39**	28	30

Table 2: Statistics of delays in days: max, mean and percentiles over all extracted trend topics for the considered approaches and matching configurations. *Config1* matches trends based on documents only (DRecall@k > 0.1); *Config2* matches trends based on keywords only (WRecall@k > 0.3 and SRecall@k > 0); *Config3* is a joint option (DRecall@k > 0.1 , WRecall@k > 0.3 and SRecall@k > 0).

BERTopic Apart from probabilistic topic models, we compared our solution to a neural-based model called BERTopic (Grootendorst, 2020) that leverages the token embeddings retrieved from BERT model (Devlin et al., 2018b). BERTopic is a topic modeling technique that uses transformers and c-TF-IDF to create dense topical clusters. First, BERTopic transforms document into embeddings. BERTopic supports many embedding models, including ones from Sentence-Transformers, Flair, Spacy, Gensim, USE. We used sentence-transformers package to get document-level embeddings. Second, BERTopic performs dimensionality reduction on the embeddings as a preparation step for clustering. Specifically, it uses UMAP (McInnes et al., 2018) as it keeps a significant portion of the high-dimensional local structure in lower dimensionality. Third, BERTopic clusters the documents with HDBSCAN (McInnes et al., 2017). Having the topical clusters, one may want to get the tokens of most importance from each cluster. Class-based TF-IDF (c-TF-IDF) is used to solve this. c-TF-IDF treats all documents in a topic as a single document and then applies TF-IDF, so that resulting TF-IDF scores demonstrate the important words in a topic.

Although BERTopic supports dynamic topic modeling, it did not fit to our purposes at all. First, BERTopic DTM creates a general topic model as if there were no temporal aspect in the documents. Then for each topic and timestep, it calculates the c-TF-IDF representation, resulting in different formulations of the same topics at different timesteps. To detect and track how new topics emerge, we trained 82 separate models, one for each timestamp respectively.

5.4 Comparison with the Baselines

We compare our solution to the aforementioned baselines using the base elements of the approach: a basic way of choosing the number of new topics and using the full history of documents for training at each step. We matched the extracted topics with the labeled trend topics using several metrics based on DRecall@k, WRecall@k and SRecall@k scores described in Section 3.3.

Three combinations of thresholds were used for matching at each timestamp:

- *Config1*: DRecall@k > 0.1, matches trends based on documents only;
- *Config2*: WRecall@k > 0.3 and SRecall@k > 0 matches trends based on keywords only;
- *Config3*: DRecall@k > 0.1 , WRecall@k > 0.3 and SRecall@k > 0 (joint option).

Table 2 shows the calculated statistics for the day delay metric. It can be seen that BERTopic model achieves the best scores for *Config1*, extracting almost all the trends in this configuration: 90 out of 91. This is due to the fact that this model has a larger number of topics and it is able to successfully distinguish documents among them. However, BERTopic is very bad at keywords extraction, since this is not its primary purpose. Therefore, for the other two configurations, its quality is much worse.

If we compare only topic models, then there is no single approach that stands out. From the table, we can conclude that PLSA is not the best choice for our task. The LDA model seems more apposite for

Config2, but ARTM is better in terms of *Config1* and *Config3*. The ARTM model generates the correct topics quickly enough even with the rigid configuration *Config3* compared to other topic models, although it can sometimes extract fewer trends in total. In the configuration *Config1*, when the main goal is to correctly divide documents by topics, ARTM extracts almost half of the trends in the first two months. Thus, it is well suited for qualitative identification of trends in the problem of early detection.

It is also worth noting that the BERTopic and ARTM models are able to extract a trend right at the moment of its inception for *Config1* (zero values for the "min" row). This is due to the fact that some of the trends are tasks in which there is no clear first paper.

To analyze the evolution of the quality metric depending on time, we explored the dependence of the proportion of detected trends from the time elapsed since their inception.

Figure 4 demonstrates the corresponding results for *Config1*. It can be used to rank models by quality in terms of document evaluation. In this case, the BERTopic model is superior to others at each timestamp, while PLSA is inferior to others. Further, ARTM is better than LDA because it extracts trends faster, although it compares later in total.

Figure 5 shows similar results for *Config2* and analyzes the quality in terms of the ranked keywords. In this case, as it was shown earlier, the BERTopic model performs much worse than the aforementioned topic models and extracts much fewer trends at any given timestamp. The quality for topic models increases approximately to the same extent. The LDA model has a slight advantage in the first months, but after a year and a half, the PLSA and ARTM models occasionally overtake it. These conclusions are consistent with Table 2.

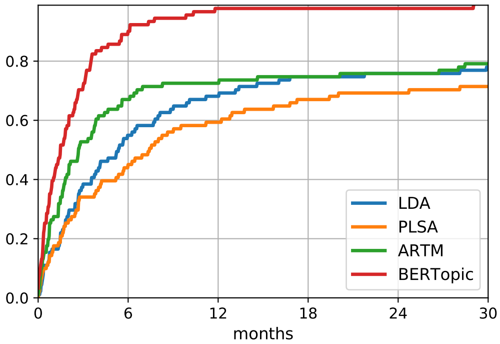

Figure 4: The dependence of the proportion of extracted trends on the months since their inception for *Config1* (DRecall@k > 0.1).

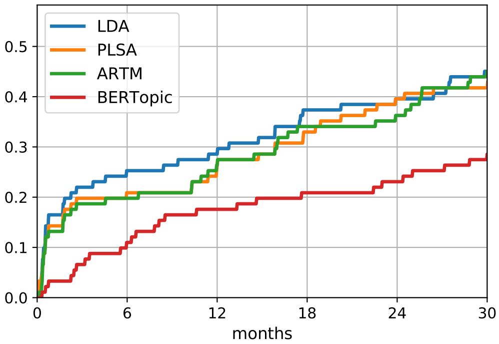

Figure 5: The dependence of the proportion of extracted trends on the months since their inception for *Config2* (WRecall@k > 0.3 and SRecall@k > 0).

In general, LDA was expected to perform better in some cases (e.g. *Config2*) because it considers the sparsity of the matrix Θ. Thus, we conducted experiments to integrate this into the model as one way of the base model modification.

Further, it should be emphasized that topic models require much less training time compared to BERTopic even though the latter is trained on GPU.

We tried to analyze why models extract some trends too late (after more than 2000 days) or not at all in some cases. Generally, the quality is limited by several factors: the sizes of topics and their presence in the validation dataset (for instance, "EM-algorithm" and "pattern recognition" present quite weakly); the occurrence of keywords in articles (the keyword "GPT" usually appears in a paper only a couple of times); the quality of the dataset and internal components of the approach (e.g. the matching procedure).

5.5 Approach Customization

Incremental Dataset In our approach, we have several options for choosing a dataset for retraining at each step. Experiments were conducted for two possible extremes (the first two options from 3): training

Statistic	*Config1*		*Config2*		*Config3*	
	B	***I***	***B***	***I***	***B***	***I***
mean	181	**67**	586	**576**	548	**498**
min	0	0	4	4	11	**4**
25%	22	**16**	**52**	214	110	**79**
50%	56	**41**	476	**452**	479	**443**
75%	120	**81**	867	**841**	**761**	793
max	3433	**514**	2711	**1921**	1949	1949
# extracted	74	**85**	53	**58**	30	**33**

Table 3: Statistics of delays in days for incremental and non-incremental dataset options. *B* denotes the base non-incremental approach (ARTM) and *I* denotes the incremental one (ARTMi). *Config1* matches trends based on documents only (DRecall@k > 0.1); *Config2* matches trends based on keywords only (WRecall@k > 0.3 and SRecall@k > 0); *Config3* is a joint option (DRecall@k > 0.1 , WRecall@k > 0.3 and SRecall@k > 0).

on the whole document history and training on the new ones only (incrementally).

Table 3 demonstrates results for ARTM. We denoted the incrementally trained approach as ARTMi. Firstly, ARTMi extracts more trends in total than ARTM in all matching configurations. For *Config1*, it is significantly superior to the LDA model and close to BERTopic. At the same time, statistics on the delay in days for it is also less than for ARTM almost in all cases. For instance, the number of days required for trend detection has decreased by almost 10 percent compared to the base model in the configuration *Config3*.

Algorithm Complexity ARTMi as well as ARTM is trained on CPU. We used Intel(R) Xeon(R) Gold 6348 CPU (24-cores) to train both ARTM and ARTMi. ARTMi can be trained in approximately 40 minutes using 16 cores in parallel. We also compared the training time for ARTM and ARTMi and found that the ARTMi model can be trained about 50 times faster. This result can be also confirmed analytically. The models take 99 batches of approximately the same size as an input. Each model runs an average of 5 times for each batch. Thus, we get $5 \cdot 99 = 495$ passes for ARTMi. The ARTM model overlaps over all previous batches at each new step. Thus, we get $5 \cdot \sum_{n=18}^{99} n = 5 \cdot 4797$ passes for ARTM, that is, 50 times more. Thus, training on an incremental dataset helps ARTMi to extract more trends in total. ARTMi is much faster than ARTM and can be effectively applied in real time. We use the incremental dataset in all the further modifications.

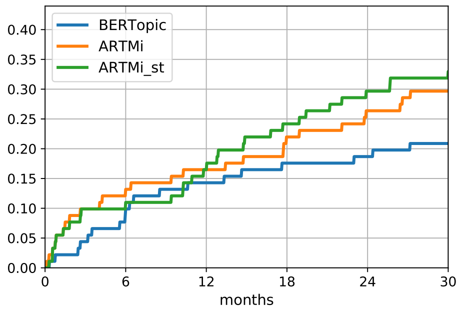

Figure 6: The dependence of the proportion of extracted trends on the months since their inception for *Config3* (DRecall@k > 0.1 , WRecall@k > 0.3 and SRecall@k > 0).

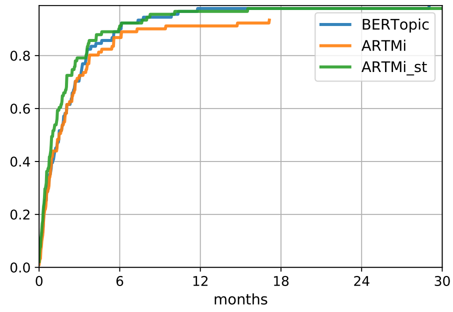

Figure 7: The dependence of the proportion of extracted trends on the months since their inception for *Config1* (DRecall@k > 0.1).

Statistic	*Config1*		*Config2*		*Config3*	
	B	***C***	***B***	***C***	***B***	***C***
mean	68	**65**	**584**	604	**554**	626
min	0	0	9	9	**9**	17
25%	12	**11**	**302**	303	256	**186**
50%	30	30	**510**	533	475	**457**
75%	74	**62**	906	**900**	**906**	912
max	949	**942**	1921	1921	**1921**	2187
# extracted	90	90	56	**57**	**40**	39

Table 4: Statistics of delays in days for two options of the number of new topics selection: *B* denotes the base approach and *C* – customized. *Config1* matches trends based on documents only (DRecall@k > 0.1); *Config2* matches trends based on keywords only (WRecall@k > 0.3 and SRecall@k > 0); *Config3* is a joint option (DRecall@k > 0.1 , WRecall@k > 0.3 and SRecall@k > 0).

Sparsity of Matrix Θ As described in Section 5.4, we have added the Θ matrix sparsity regularizer to the standard ARTM model. We denoted this model as ARTMi_st. BERTopic was also used for comparison since it: *(1)* is different from the topic models in substance and does not have any regularizations; *(2)* obtained the best results for *Config1*.

Figure 6 shows the dependence of the proportion of extracted trends on the months since their inception for the balanced configuration *Config3*. It can be seen that ARTMi_st outperforms both BERTopic and ARTMi almost for all the timestamps. It is able to extract more trends more quickly even for complex matching, evaluating both documents and keywords. In total, ARTMi_st extracted 40 trends, whereas ARTM and BERTopic – only 33 and 28 respectively.

Moreover, in the first months, the ARTMi_st model immediately overtakes BERTopic (Fig. 7, *Config1*), despite the fact that the latter outperformed the base models by a large margin. Therefore, even for distinguishing documents by topics, the topic model with decorrelation and regularization performs better than the neural BERT-based approach. Thus, adding Θ sparsity regularizer is one of the crucial components of the topic model to achieve the best quality.

Number of New Topics We experimented with two ways of the number of new topics selection (described in Section 3.2) for the ARTMi_st model. The results are demonstrated in Table 4. For *Config1*, the customized option is better than the base one. It extracts the same amount of trends in total, but it does so earlier in time. For the matching configurations associated with the presence of the correct keywords, the results are about the same as in the base case. The number of extracted topics differs by one, and the difference between delays in days is not statistically significant. Thus, the customized way of choosing the number of new topics improves the quality for some matching configurations, but it does not provide significant advantages for others.

6 Future Work

Firstly, we highlight a direction related to the trend identification subtask. We are going to leverage the document-topic distribution matrix to construct trend profiles in time. Such profiles will allow us to track the evolution of topics over time and, in case of exponential growth, serve as one of the trend indicators. Secondly, we are going to analyze and visualize the current results of the early trend detection. Besides, we plan to apply the proposed approach to other scientific areas except for machine learning and AI.

Possible applications of the proposed technology include news monitoring and extraction of the most relevant trends in different domains, assistance with scientific research (e.g. automatic tracking of emerging topics of interest) and help with literature review composing. Furthermore, such a technique may appear useful not only in scientific or news monitoring areas, but also in corporate segment for structuring and analysing large piles of legal documentation and technical requirements.

7 Conclusion

In this paper, we investigated the topic modeling approaches to the scientific trend topics detection task. The main goal was to make predictions in real-time. To this end, we customized the standard ARTM-based approach and proposed incremental training consisting of incremental initialization, incremental dataset and the number of topics updating based on the current vocabulary of trend collocations. Apart from that, we integrated sparsity regularization into our approach which increased the model quality. Our method is universal and is not model-specific.

We described the validation process and proposed a method for matching labeled trends and extracted topics. We collected the expertly labeled specialized dataset, namely AITD, to validate approaches solving early trend topic detection task. The dataset consists of 91 groups of machine learning and AI articles (each group corresponds to one trend topic) with corresponding keywords selected from publications from top conferences and alternative trend names.

The evaluation demonstrated that the basic ARTM model achieves one of the best results compared to the other baselines using different matching configurations. Moreover, incremental training techniques and additional regularization led to a significant improve in the base model quality regarding early trend detection. The final ARTM-based approach extracts the largest number of trends at the early stages of their evolution, and can operate in real-time since it requires the least training time.

References

Dimo Angelov. 2020. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*.

Sahar Behpour, Mohammadmahdi Mohammadi, Mark V. Albert, Zinat S. Alam, Lingling Wang, and Ting Xiao. 2021. Automatic trend detection: Time-biased document clustering. *Knowledge-Based Systems*, 220:106907.

Xin Bing, Florentina Bunea, and Marten Wegkamp. 2020. A fast algorithm with minimax optimal guarantees for topic models with an unknown number of topics.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Kevin Canini, Lei Shi, and Thomas Griffiths. 2009. Online inference of topics with latent dirichlet allocation. *Journal of Machine Learning Research - Proceedings Track*, 5:65–72, 01.

Jonathan Chang and David M. Blei. 2010. Hierarchical relational models for document networks. *The Annals of Applied Statistics*, 4(1), Mar.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *ArXiv*, abs/2003.10555.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2019. The dynamic embedded topic model. *arXiv preprint arXiv:1907.05545*.

Adji Bousso Dieng. 2020. *Deep Probabilistic Graphical Modeling*. Columbia University.

Ahmed El-Kishky, Yanglei Song, Chi Wang, Clare R. Voss, and Jiawei Han. 2014. Scalable topical phrase mining from text corpora. *ArXiv*, abs/1406.6312.

C. Erten, P. J. Harding, S. G. Kobourov, K. Wampler, and G. Yee. 2004. Exploring the computing literature using temporal graph visualization. *Proceedings of SPIE - The International Society for Optical Engineering*, 5295:45–56. Visualization and Data Analysis 2004 ; Conference date: 19-01-2004 Through 20-01-2004.

Michael Färber and Adam Jatowt. 2019. Finding temporal trends of scientific concepts. *// BIR@ ECIR*, P 132–139.

Maarten Grootendorst. 2020. Bertopic: Leveraging bert and c-tf-idf to create easily interpretable topics.

Minor Eduardo Quesada Grosso, Edgar Casasola, and Jorge Antonio Leoni de León. 2017. Trending topic extraction using topic models and biterm discrimination. *CLEI Electron. J.*, 20(1):3:1–3:13.

Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, and Lee Giles. 2009. Detecting topic evolution in scientific literature: How can citations help? // *ACM 18th International Conference on Information and Knowledge Management, CIKM 2009*, International Conference on Information and Knowledge Management, Proceedings, P 957–966. ACM 18th International Conference on Information and Knowledge Management, CIKM 2009 ; Conference date: 02-11-2009 Through 06-11-2009.

Jonathan C. Ho, Ewe-Chai Saw, Louis Y.Y. Lu, and John S. Liu. 2014. Technological barriers and research trends in fuel cell technologies: A citation network analysis. *Technological Forecasting and Social Change*, 82(C):66–79.

Matthew Hoffman, David Blei, and Francis Bach. 2010. Online learning for latent dirichlet allocation. volume 23, P 856–864, 11.

Thomas Hoffman. 1999. Probabilistic latent semantic indexing. // *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval, 1999*, P 50–57.

Anastasia Ianina and Konstantin Vorontsov. 2019. Regularized multimodal hierarchical topic model for document-by-document exploratory search. // *2019 25th Conference of Open Innovations Association (FRUCT)*, P 131–138. IEEE.

Anastasia Ianina and Konstantin Vorontsov. 2020. Hierarchical interpretable topical embeddings for exploratory search and real-time document tracking. *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*, 11(4):134–152.

Yookyung Jo, Carl Lagoze, and C. Lee Giles. 2007. Detecting research topics via the correlation between graphs and texts. // *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, P 370–379, New York, NY, USA. Association for Computing Machinery.

April Kontostathis, Leon M. Galitsky, William M. Pottenger, Soma Roy, and Daniel J. Phelps, 2004. *A Survey of Emerging Trend Detection in Textual Data Mining*, P 185–224. Springer New York, New York, NY.

Mikhail Krivenko and Vitaly Vasilyev. 2009. Sequential latent semantic indexing. // *Proceedings of the 2nd Workshop on Data Mining using Matrices and Tensors*, P 1–9.

Minh-Hoang Le, Tu Bao Ho, and Yoshiteru Nakamori. 2006. Detecting emerging trends from scientific corpora.

June Young Lee, Sejung Ahn, and Dohyun Kim. 2021. Deep learning-based prediction of future growth potential of technologies. *Plos one*, 16(6):e0252753.

Dingge Liang, Marco Corneli, Charles Bouveyron, and Pierre Latouche. 2023. The graph embedded topic model.

Manuel V Loureiro, Steven Derby, and Tri Kurniawan Wijaya. 2023. Topics as entity clusters: Entity-based topics from language models and graph neural networks. *arXiv preprint arXiv:2301.02458.*

Huifang Ma, Zhixin Li, and Zhongzhi Shi. 2010. Combining the missing link: An incremental topic model of document content and hyperlink. // Zhongzhi Shi, Sunil Vadera, Agnar Aamodt, and David B. Leake, *Intelligent Information Processing V - 6th IFIP TC 12 International Conference, IIP 2010, Manchester, UK, October 13-16, 2010. Proceedings*, volume 340 of *IFIP Advances in Information and Communication Technology*, P 259–270. Springer.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426.*

Qiaozhu Mei and ChengXiang Zhai. 2005. Discovering evolutionary theme patterns from text - an exploration of temporal text mining. P 198–207, 01.

Alena Moiseeva and Hinrich Schütze. 2020. Trendnert: A benchmark for trend and downtrend detection in a scientific domain. // *AAAI.*

Vinodkumar Prabhakaran, William L Hamilton, Dan McFarland, and Dan Jurafsky. 2016. Predicting the rise and fall of scientific topics from trends in their rhetorical framing. // *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, P 1170–1180.

Hamed Rahimi, Hubert Naacke, Camelia Constantin, and Bernd Amann. 2023. Antm: An aligned neural topic model for exploring evolving topics. *arXiv preprint arXiv:2302.01501*.

Daniele Rotolo, Diana Hicks, and Ben R. Martin. 2015. What is an emerging technology? *Research Policy*, 44(10):1827–1843.

Soma Roy, David Gevry, and William Pottenger. 2002. Methodologies for trend detection in textual data mining. 2, 10.

S Sivanandham, A Sathish Kumar, R Pradeep, and Rajeswari Sridhar. 2021. Analysing research trends using topic modelling and trend prediction. // *Soft Computing and Signal Processing: Proceedings of 3rd ICSCSP 2020, Volume 1*, P 157–166. Springer.

Ana Sabina Uban, Cornelia Caragea, and Liviu P Dinu. 2021. Studying the evolution of scientific topics and their relationships. // *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, P 1908–1922.

Konstantin Vorontsov and Anna Potapenko. 2015. Additive regularization of topic models. *Machine Learning*, 101(1):303–323.

Konstantin Vorontsov, Oleksandr Frei, Murat Apishev, Peter Romov, and Marina Dudarenko. 2015. Bigartm: Open source library for regularized multimodal topic modeling of large collections. // *International Conference on Analysis of Images, Social Networks and Texts*, P 370–381. Springer.

Delvin Ce Zhang and Hady Lauw. 2022. Dynamic topic models for temporal document networks. // *International Conference on Machine Learning*, P 26281–26292. PMLR.